UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/762,804	01/22/2004	Yi-Min Wang	124439.09	5395

22971 7590 02/05/2009

MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052-6399

EXAMINER	
ZHEN, LI B	
ART UNIT	PAPER NUMBER
2194	

NOTIFICATION DATE	DELIVERY MODE
02/05/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

roks@microsoft.com
ntovar@microsoft.com

Office Action Summary	Application No.	Applicant(s)	
	10/762,804	WANG ET AL.	
	Examiner	Art Unit	
	LI B. ZHEN	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 June 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *61-118* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *61-118* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *1/22/2004*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Preliminary amendment filed 01/22/2004, cancels claims 1 – 60 and adds new claims 61 - 118. Claims 61 - 118 are presented for examination.

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted on 1/24/2004 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is considered by the examiner.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. **Claims 61 – 118 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1 – 58 of U.S. Patent No. 6,708,223 [hereinafter Patent223] in view of "DCOM and CORBA Side by Side, Step by Step, and Layer by Layer" [hereinafter Chung].**

In determining whether a nonstatutory basis exists for a double patenting rejection, the first question to be asked is — does any claim in the application define an invention that is anticipated by, or is merely an obvious variation of, an invention claimed in the patent? If the answer is yes, then an "obviousness-type" nonstatutory double patenting rejection may be appropriate. See MPEP 804 (II)(B)(1) Nonstatutory Double Patenting, Obvious-Type. The difference between the inventions defined by the conflicting claims are as follows: **(1)** Patent223 specify the objects a Distributed Component Object Models (DCOM) objects and interface pointer identifiers as DCOM interface pointer identifiers; and **(2)** current application includes the additional feature of a bypassed mechanism that adds a remote procedure call interface identifier to the call (claims 61, 76 and 118) or a remote procedure call utility layer comprising a pointer to the dispatching function that allows the call to be passed directly to the dispatching function (claims 91, 104 and 117).

As to difference (1), the DCOM objects and interface pointer identifiers corresponds to the objects and interface pointer identifiers claimed in the current application.

As to difference (2), Chung teaches disclose the calling the interface of the second object by the first object comprises bypassing a mechanism [DCOM also

provides a custom marshaling mechanism to bypass the standard marshaling procedure; Section 4, p. 10], the bypassed mechanism comprising adding a remote procedure call interface identifier to the call [object stub marshals the interface pointer; p. 6, right column, step 6], the remote procedure call utility functions are performed on the received call by a remote procedure call utility layer [middle layer, p. 3, 1st paragraph of Chung], the remote procedure call utility layer comprising a pointer to the dispatching function [object stub marshals the interface pointer; p. 6, right column, step 6 of Chung], wherein the pointer allows the call to be passed directly to the dispatching layer [DCOM also provides a custom marshaling mechanism to bypass the standard marshaling procedure; Section 4, p. 10 of Chung].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the invention as disclosed in Patent223 to incorporate the features of Chung. One of ordinary skill in the art would have been motivated to make the combination because this allows a server object to declare that it wants to control how and what data are marshaled and unmarshaled, and how the client should communicate with the server [p. 10, Section 4].

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7. **Claims 61, 76, 91, 104, 117 and 118 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Harnessing User-Level Networking Architectures for Distributed Object Computing over High-Speed Networks" [cited in IDS dated 1/22/2004, hereinafter Madu] in view of "DCOM and CORBA Side by Side, Step by Step, and Layer by Layer" [hereinafter Chung].**

8. As to claim 61, Madu teaches a method of communication [p. 4, Section 4. DCOM Remote Method Invocation over VI Architecture Transport] between a first object located [custom proxy] on a first computer and a second object [custom stub] located on a second computer, the first and second computers connected by a network [VI Architecture is a user-level networking architecture, Section 2. Virtual Interface Architecture, p. 2], the method comprising:

calling an interface of the second object by the first object on the first computer [user-level VI transport for inter-process communications, Fig. 5], wherein the interface

of the second object is identified only with an interface pointer identifier [interface pointers in COM are passed as parameters in method calls, Section 3.1, p.3];

performing remote procedure call utility functions on the call at the first computer [custom marshalling architecture that uses the high performance user-level VI transport for inter-process communication, Section 4, p. 4]; and

communicating the call to the second computer, wherein the second computer receives the call [Section 4.2], performs remote procedure call utility functions on the call [custom stub manager manages endpoint creation and destruction, data transfers, object lifetime, and dispatches method requests to interface stubs; Section 4.2], passes the call to a dispatching function so as to bypass a remote procedure call dispatching function [see Fig. 5, standard RPC is bypassed], invokes a stub [Section 4, method invocation], and accesses the interface of the second object identified by the interface pointer identifier [interface stub unmarshals method parameters and dispatches actual object methods, Section 4.2, p. 5]. Madu does not specifically disclose the calling the interface of the second object by the first object comprises bypassing a mechanism, the bypassed mechanism comprising adding a remote procedure call interface identifier to the call.

However, Chung teaches disclose the calling the interface of the second object by the first object comprises bypassing a mechanism [DCOM also provides a custom marshaling mechanism to bypass the standard marshaling procedure; Section 4, p. 10], the bypassed mechanism comprising adding a remote procedure call interface identifier to the call [object stub marshals the interface pointer; p. 6, right column, step 6].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the invention of Madu to incorporate the features of Chung. One of ordinary skill in the art would have been motivated to make the combination because this allows a server object to declare that it wants to control how and what data are marshaled and unmarshaled, and how the client should communicate with the server [p. 10, Section 4].

9. As to claim 76, this is a program product claim that corresponds to method claim 61; see the rejection to claim 61 above, which also meet the limitations of this program product claim.

10. As to claim 91, Madu as modified teaches a method of communication between a first object located on a first computer and a second object located on a second computer [p. 4, Section 4 of Madu], the first and second computers connected by a network [Section 2. Virtual Interface Architecture, p. 2 of Madu], the method comprising:

receiving, at the second computer [Section 4.2 of Madu], a call to an interface of the second object from the first object on the first computer [user-level VI transport for inter-process communications, Fig. 5 of Madu], wherein the interface of the second object is identified only with an interface pointer identifier [interface pointers in COM are passed as parameters in method calls, Section 3.1, p.3 of Madu];

performing remote procedure call utility functions on the received call [custom marshalling…for inter-process communication, Section 4, p. 4 of Madu], wherein the

remote procedure call utility functions are performed on the received call by a remote procedure call utility layer [middle layer, p. 3, 1st paragraph of Chung], the remote procedure call utility layer comprising a pointer to the dispatching function [object stub marshals the interface pointer; p. 6, right column, step 6 of Chung], wherein the pointer allows the call to be passed directly to the dispatching layer [DCOM also provides a custom marshaling mechanism to bypass the standard marshaling procedure; Section 4, p. 10 of Chung];

passing the received call to a dispatching function so as to bypass a remote procedure call dispatching function [see Fig. 5, standard RPC is bypassed of Madu];

invoking a stub [Section 4, method invocation of Madu]; and

accessing the interface of the second object identified by the interface pointer identifier [Section 4.2, p. 5 of Madu].

11. As to claim 104, this is a program product claim that corresponds to method claim 91; see the rejection to claim 104 above, which also meet the limitations of this program product claim.

12. As to claim 117, Madu as modified a computing device [p. 5, Section 4.2 of Madu] comprising:

an object [server object, p. 4, Section 4 of Madu], the object comprising an interface that is called by a second object [client object, p. 4, Section 4 of Madu] on a

second computing device, wherein the interface is identified only with an interface pointer identifier [Section 3.1, p.3 of Madu];

a network connection, wherein the network connection communicationally connects the computing device to the second computing device [Section 2. Virtual Interface Architecture, p. 2 of Madu];

a remote procedure call utility layer [custom stub manager, Section 4.2 of Madu], wherein the remote procedure call utility layer performs remote procedure call utility functions on the interface call by the second object [Section 4, p. 4 of Madu] and passes the interface call to a dispatching function so as to bypass a remote procedure call dispatching function [see Fig. 5, standard RPC is bypassed of Madu] and wherein the remote procedure call utility layer comprises a pointer to the dispatching function [p. 6, right column, step 6 of Chung], wherein the pointer allows the call to be passed directly to the dispatching function [Section 4, p. 10 of Chung]; and

a dispatching layer comprising the dispatching function, wherein the dispatching layer invokes a stub [Section 4, method invocation of Madu] and accesses the interface identified by the interface pointer identifier [Section 4.2, p. 5 of Madu].

13. As to claim 118, Madu as modified teaches a computing device [p. 5, Section 4.2 of Madu] comprising:

an object [server object, p. 4, Section 4 of Madu], the object calling an interface of a second object on a second computing device [client object, p. 4, Section 4 of

Madu], wherein the interface is identified only with an interface pointer identifier [Section 3.1, p.3 of Madu];

a remote procedure call utility layer [custom stub manager, Section 4.2 of Madu], wherein the remote procedure call utility layer performs remote procedure call utility functions on the call [Section 4, p. 4 of Madu];

a bypass of a mechanism [Section 4, p. 10 of Chung], the mechanism comprising adding a remote procedure call interface identifier to the call [p. 6, right column, step 6 of Chung]; and

a network connection, wherein the network connection communicates the call to the second computing device, and wherein further the second computing device receives the call [Section 2. Virtual Interface Architecture, p. 2 of Madu], performs remote procedure call utility functions on the call [Section 4, p. 4 of Madu], passes the call to a dispatching function so as to bypass a remote procedure call dispatching function [see Fig. 5, standard RPC is bypassed of Madu], invokes a stub [Section 4, method invocation of Madu], and accesses the interface of the second object identified by the interface pointer identifier [Section 4.2, p. 5 of Madu].

14. **Claims 62, 65 – 67, 69, 70, 72, 75, 77, 80 – 82, 84, 85, 87, 90, 92, 93, 96 – 101, 105, 106 and 109 – 114 are rejected under 35 U.S.C. 103(a) as being unpatentable over Madu and Chung further in view of "Virtual Interface Architecture Specification, Revision 1.0" [cited in IDS dated 1/22/2004, hereinafter VIA] and U.S. Patent No. 6,044,409 to Lim.**

15. As to claims 62 and 77, Madu teaches a first send buffer [send queue, p. 2, Section 2], a first receive buffer [receive queue, p. 2, Section 2]. Madu does not teach the first receive buffer is posted to be of sufficient size to accept the second data.

However, VIA teaches a first receive buffer [VI Consumer at the receiving end pre-posts a Descriptor to the receive queue, first paragraph, p. 15 of VIA], and the first receive buffer is posted to be of sufficient size to accept the second data [VI Consumer on the receiving side must post a Receive Descriptor of sufficient size before the sender's data arrives, second full paragraph, p. 15 of VIA].

It would have been obvious to apply the teaching of creating a receiver buffer that is of sufficient size to accept data as taught VIA to the invention of Madu because creating a buffer that is sufficient to accept incoming data would prevent buffer overflow.

Madu as modified does not specify posting on the first computer a first receive buffer prior to sending a first data to the second computer.

However, Lim teaches [column 12, lines 19 – 25, 55 – 60, and 64 – 66] posting on the first computer a first receive buffer prior to sending a first data to the second computer [a marshal buffer appropriate for the transport selected is created in step 206, Fig. 4], the first receive buffer will receive a second data from the second computer in response to the first data [the client receives a reply from the server and encapsulates the reply in a marshal buffer 216 and 218, Fig. 4], and sending the first data to the

second computer [the contents of the marshal buffer are transmitted over the selected transport to the identified end point 212, Fig. 4].

It would have been obvious to apply the teaching of posting on the first computer a first receive buffer prior to sending a first data to the second computer as taught by Lim to the invention of Madu as modified because this would ensure that there is memory available to store the response data.

16. As to claims 65, 75, 80 and 90, Madu as modified teaches the second data from the second computer is in response to the first data from the first computer [the client receives a reply from the server and encapsulates the reply in a marshal buffer 216 and 218, Fig. 4; column 12, lines 19 – 25, 55 – 60, and 64 – 66 of Lim].

17. As to claims 66 and 81, Madu as modified teaches the first computer has a first memory location and a buffer [p. 12 – 13, Section 2.1.1. Virtual Interfaces of VIA], and access to the network through an interface card on the first computer [VI NIC; p. 12 – 13, Section 2.1.1 of VIA], the method further comprising: placing in the buffer a copy of a first pointer to a first parameter [Descriptor is a data structure that contains all of the information that the VI Provider needs to process the request, such as pointers to the data buffers; p. 12 – 13, Section 2.1.1 of VIA], wherein the first parameter is used in the calling of the interface of the second object [p. 5, Section 4.2 Anatomy of Custom Stub/Proxy; p. 2, Section 2. Virtual Interface Architecture of Madu] and wherein the first pointer points to the first parameter in the first memory location [receive queue contains

Descriptors that describe where to place incoming data, p. 10 of VIA]; and transmitting, by the network interface card, the first parameter pointed to by the first pointer by reading the first parameter out of the first memory location [VI NIC directly performs data transfer functions; p. 12 – 13, Section 2.1.1 of VIA].

18. As to claims 67 and 82, Madu as modified teaches issuing a notification on the first computer after the network interface card has finished reading the first parameter out of the first memory location [p. 15, first paragraph of VIA].

19. As to claims 69 and 84, Madu as modified teaches placing in the buffer a copy of the first pointer to the first parameter and a copy of a second pointer to a second parameter [p. 12 – 13, Section 2.1.1. Virtual Interfaces of VIA, wherein the second parameter is used in the calling of the interface of the second object and wherein the second pointer points to the second parameter in a second memory location on the first computer [p. 5, Section 4.2 Anatomy of Custom Stub/Proxy; p. 2, Section 2. Virtual Interface Architecture of Madu]; and transmitting, by the network interface card, the first parameter pointed to by the first pointer by reading the parameter out of the first memory location and the second parameter pointed to by the second pointer by reading the second parameter out of the second memory location [VI NIC directly performs data transfer functions; p. 12 – 13, Section 2.1.1 of VIA].

20. As to claims 70 and 85, Madu as modified teaches issuing a first notification on the first computer after the network interface card has finished reading the first parameter out of the first memory location and issuing a second notification on the first computer after the network interface card has finished reading the second parameter out of the second memory location [p. 15, first paragraph of VIA].

21. As to claims 72 and 87, Madu as modified teaches posting, on the first computer, a first send buffer and a first receive buffer prior to sending a first data to the second computer [column 12, lines 19 – 25, 55 – 60, and 64 – 66 of Lim], wherein the first receive buffer will receive a second data from the second computer [client receives a reply from the server; column 12, lines 19 – 25, 55 – 60, and 64 – 66 of Lim], and wherein the first receive buffer is posted to be of sufficient size to accept the second data [p. 15 of VIA]; and sending the first data to the second computer via the first send buffer [the contents of the marshal buffer are transmitted over the selected transport to the identified end point 212, Fig. 4; column 12, lines 19 – 25, 55 – 60, and 64 – 66 of Lim].

22. As to claims 92 and 105, Madu as modified teaches storing, on the second computer, a second data into a first receive buffer [Section 4.2, p. 5 of Madu], wherein the first receive buffer was posted prior to sending a first data to the first computer [column 12, lines 19 – 25, 55 – 60, and 64 – 66 of Lim].

23. As to claims 93 and 106, Madu as modified teaches the first data to the first computer was sent prior to receiving the second data from the first computer [column 12, lines 55 – 67 of Lim].

24. As to claims 96 and 109, Madu as modified teaches the second computer has a memory storage location and a buffer [Section 4.2, p. 5 of Madu], and access to the network through a network interface card on the second computer [p. 15, first paragraph of VIA], the method further comprising: receiving a call from the first object on the interface of the second object [p. 12 – 13, Section 2.1.1 of VIA], receiving, by the network interface card, a parameter of the call from the first object [VI NIC directly performs data transfer functions; p. 12 – 13, Section 2.1.1 of VIA]; storing the parameter in a memory location [receive queue contains Descriptors that describe where to place incoming data, p. 10 of VIA]; and accessing, by the second object, the parameter [p. 5, Section 4.2 of Madu].

25. As to claims 97 and 110, Madu as modified teaches the memory location is the buffer, and wherein the accessing the parameter is performed in the buffer [p. 5, Section 4.2 of Madu].

26. As to claim 98 and 111, Madu as modified teaches copying the parameter from the buffer into the memory storage location, wherein the accessing the parameter is performed in the memory storage location [p. 12 – 13, Section 2.1.1 of VIA].

27. As to claims 99 and 112, Madu as modified teaches the memory location is the memory storage location, and wherein the accessing the parameter is performed in the memory storage location [Section 2.2, p. 14 of VIA].

28. As to claims 100 and 113, Madu as modified teaches storing, on the second computer, a second data into a first receive buffer [p. 15 of VIA], wherein the first receive buffer was posted prior to sending a first data to the first computer [column 12, lines 19 – 25, 55 – 60, and 64 – 66 of Lim], and wherein the first receive buffer was posted to be of sufficient size to accept the second data [p. 15 of VIA].

29. As to claims 101 and 114, Madu as modified teaches the first data to the first computer was sent prior to receiving the second data from the first computer [column 12, lines 55 – 67 of Lim].

30. **Claims 63, 64, 68, 71, 73, 74, 78, 79, 83, 86, 88, 89, 94, 95, 102, 103, 107, 108, 115 and 116 are rejected under 35 U.S.C. 103(a) as being unpatentable over Madu, Chang, VIA and Lim further in view of U.S. Patent No. 6,131,126 to Kougiouris.**

31. As to claims 63, 64, 78 and 79, Madu as modified does not specify reclaiming the memory location after data transmission.

However, Kougiouris teaches [column 2, lines 28 – 45] a method in a computer system for inter-process communication that reclaims a memory location after data transmission [the first buffer is deallocated upon receipt of the communication].

It would have been obvious to apply the teaching of reclaiming a memory location after data transmission as taught by Kougiouris to the invention of Madu as modified because this prevents large and unnecessary consumption of memory resources.

32. As to claims 68 and 83, Madu as modified teaches reclaiming the first memory location after receiving the notification [column 2, lines 28 – 45 of Kougiouris].

33. As to claims 71 and 86, Madu as modified teaches reclaiming the first memory location after receiving the first notification and reclaiming the second memory location after receiving the second notification [column 2, lines 28 – 45 of Kougiouris].

34. As to claims 73 and 88, Madu as modified teaches cleaning up, on the first computer, a second receive buffer after sending the first data to the second computer and prior to receiving the second data from the second computer [column 2, lines 28 – 45 of Kougiouris].

35. As to claims 74 and 89, Madu as modified teaches cleaning up, on the first computer, a second send buffer after sending the first data to the second computer and

prior to receiving the second data from the second computer [column 2, lines 28 – 45 of Kougiouris].

36. As to claims 94 and 107, Madu as modified teaches cleaning up, on the second computer, a send buffer after sending the first data to the first computer and prior to receiving the second data from the first computer [column 2, lines 28 – 45 of Kougiouris].

37. As to claims 95 and 108, Madu as modified teaches cleaning up, on the second computer, a second receive buffer after sending the first data to the first computer and prior to receiving the second data from the first computer [column 2, lines 28 – 45 of Kougiouris].

38. As to claims 102 and 115, Madu as modified teaches cleaning up, on the second computer, a send buffer after sending the first data to the first computer and prior to receiving the second data from the first computer [column 2, lines 28 – 45 of Kougiouris].

39. As to claims 103 and 116, Madu as modified teaches cleaning up, on the second computer, a second receive buffer after sending the first data to the first computer and prior to receiving the second data from the first computer [column 2, lines 28 – 45 of Kougiouris].

CONTACT INFORMATION

40. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Li B. Zhen/
Primary Examiner, Art Unit 2194

Li B. Zhen
Primary Examiner
Art Unit 2194